

# Integrate Grafana with Linux Server for high cpu utilization and create a graph in Grafana.



# **Table of Content:**

## **1. What is Grafana?**

## **2. How to Install Grafana**

- Create EC2 Instance**
- Install Grafana**
- Test Grafana**

## **3. Configure AWS cloudwatch dashboard**

- Create a new Policy**
- Creating a Role**
- Creating a User**
- Attach the user to EC2 Instance**
- Create your Dashboard**

## **4. Conclusion**

## **1. What is Grafana?**

Per Grafana “Grafana is a complete observability stack that allows you to monitor and analyze metrics, logs, and traces. It allows you to query, visualize, alert on and understand your data no matter where it is stored. Create, explore, and share beautiful dashboards with your team and foster a data-driven culture”. This is a very useful tool to visualize different logs, errors, or metrics from your instances.

## **2. How To Install Grafana**

### **Step 1: Create EC2 Instance**

We need to install Grafana on our server first, so let's provision an EC2 instance

- Go to EC2
- Create EC2
- Choose a public subnet

- Give it our Grafana security group

Now SSH into our instance

- Update your packages with

```
sudo yum update -y
```

We need to add a repository for grafana so our OS will know where it is.

```
sudo nano /etc/yum.repos.d/grafana.repo
```

Add the text below to the repo file. This will install the open-source Grafana.

## Step 2: Install Grafana

Now we will install Grafana. *If you wish to actually test Grafana skip to Section Two below before you install.*

```
sudo yum install grafana -y
```

The next command will reload the system

```
sudo systemctl daemon-reload
```

Then we will start our server and check our service with the following

two commands.

```
sudo systemctl start grafana-server  
sudo systemctl status grafana-server
```

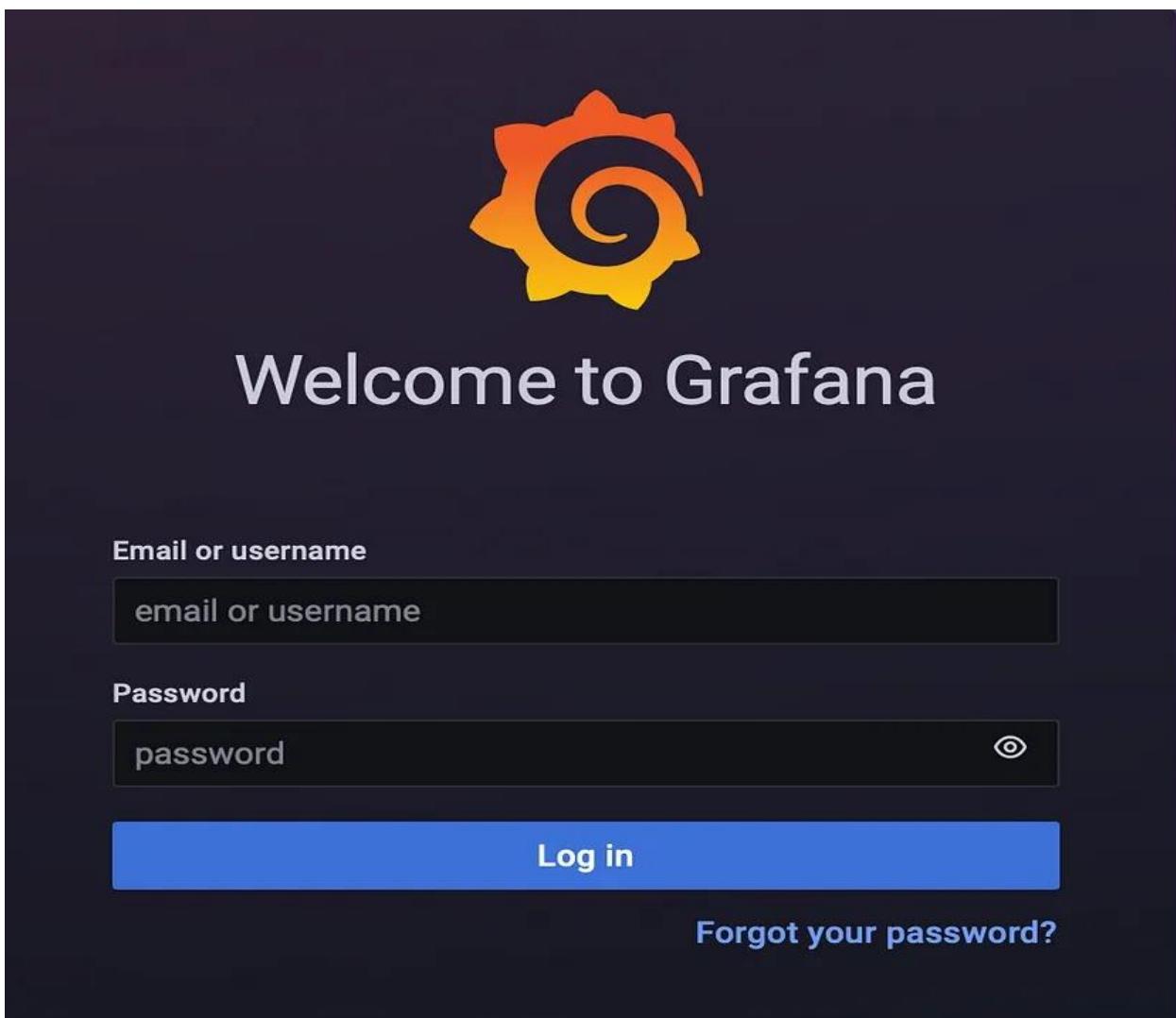
```
● grafana-server.service - Grafana instance  
  Loaded: loaded (/usr/lib/systemd/system/grafana-server.service)  
  Active: active (running) since Thu 2022-06-09 10:41:41 UTC  
            Docs: http://docs.grafana.org  
    Main PID: 6507 (grafana-server)  
      CGroup: /system.slice/grafana-server.service  
              └─6507 /usr/sbin/grafana-server --config /etc/grafana/grafana.conf
```

Our final command will ensure that Grafana will start up automatically if we stop and restart our instance.

```
sudo systemctl enable grafana-server.service
```

## Step 3:Test Grafana

To test our server we will need to grab our public IPv4 and add a :3000 at the end, ie. 10.90.80.10:3000, and insert it into our browser URL. This will bring you to a login screen and our Username and Password will be admin. You will be prompted to create a new password.



## Step 4: Create a new Policy

The screenshot shows the AWS Identity and Access Management (IAM) Policies page. The left sidebar includes links for Dashboard, Access management (User groups, Users, Roles), Policies (Identity providers, Account settings), and Access reports. A search bar for 'Search IAM' is also present. The main area has tabs for 'Create policy' (which is selected) and 'Policy actions'. A 'Filter policies' dropdown and a search bar are at the top of the policy list table. The table columns are Policy name, Type, Used as, and Actions (indicated by icons). A new policy named 'create\_policy\_IAM' is listed at the bottom of the table.

	Policy name	Type	Used as	A
...	AccessAnalyzerServiceRole...	AWS managed	None	A
...	AdministratorAccess	Job function	None	P
...	AdministratorAccess-Amplify	AWS managed	None	G
...	AdministratorAccess-AWSEI...	AWS managed	None	G
...	AlexaForBusinessDeviceSe...	AWS managed	None	P
...	AlexaForBusinessFullAccess	AWS managed	None	G
...	AlexaForBusinessGateway...	AWS managed	None	P
...	AlexaForBusinessLifesizeD...	AWS managed	None	P
...	AlexaForBusinessNatgeoD...	AWS managed	None	T
	create_policy_IAM			

Just follow this, under **Services** → **select IAM** → **Policies** →  
**Create Policy**

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

Visual editor    JSON    Import managed policy

Expand all | Collapse all

Select a service

Service Select a service below

close

Access Analyzer EI MediaTailor  
Account EKS MGN  
Activate Elastic Beanstalk Migration Hub  
Alexa for Business Elastic Block Store Mobile Analytics  
AMG Elastic Container Registry Mobile Hub

choose\_service

This screenshot shows the AWS IAM Policy Visual Editor interface. At the top, there are tabs for 'Visual editor' (which is selected) and 'JSON'. To the right, there's a link to 'Import managed policy'. Below the tabs, there are buttons for 'Expand all' and 'Collapse all'. A search bar labeled 'Find a service' is present. The main area is titled 'Select a service' and contains a list of services under the heading 'Service'. The services listed are: Access Analyzer, EI, MediaTailor; Account, EKS, MGN; Activate, Elastic Beanstalk, Migration Hub; Alexa for Business, Elastic Block Store, Mobile Analytics; and AMG, Elastic Container Registry, Mobile Hub. There are also 'Clone' and 'Remove' buttons at the top right. A note 'Enter service manually' is visible. The entire interface is enclosed in a light gray border.

Now under the services → Cloud Watch

## Step 5: Creating a Role

Identity and Access Management (IAM)

Create role    Delete role

Dashboard

Access management

User groups

Users

**Roles**

Policies

Identity providers

Account settings

Access reports

Search IAM

Role name   Trusted entities   Last ac...

Role name	Trusted entities	Last ac...
AmazonSageMaker-ExecutionRole-20200...	AWS service: sagemaker	293 day
AWSServiceRoleForIoTSiteWise	AWS service: iotsitewise (Service-Linked role)	None
AWSServiceRoleForLexBots	AWS service: lex (Service-Linked role)	None
AWSServiceRoleForRDS	AWS service: rds (Service-Linked role)	177 day
AWSServiceRoleForSupport	AWS service: support (Service-Linked role)	None
AWSServiceRoleForTrustedAdvisor	AWS service: trustedadvisor (Service-Linked ...)	None
localization_role	AWS service: lambda	None
localization-test	AWS service: lambda	155 day
narenlk_csv_dynamodb	AWS service: lambda	156 day

create\_roles

This screenshot shows the AWS IAM Roles page. On the left, there's a sidebar with links for 'Dashboard', 'Access management', 'User groups', 'Users', 'Roles' (which is highlighted in orange), 'Policies', 'Identity providers', 'Account settings', and 'Access reports'. Below the sidebar is a search bar labeled 'Search IAM'. The main area has buttons for 'Create role' and 'Delete role'. A table lists existing roles based on their 'Role name', 'Trusted entities', and 'Last ac...'. The roles listed include 'AmazonSageMaker-ExecutionRole-20200...', 'AWSServiceRoleForIoTSiteWise', 'AWSServiceRoleForLexBots', 'AWSServiceRoleForRDS', 'AWSServiceRoleForSupport', 'AWSServiceRoleForTrustedAdvisor', 'localization\_role', 'localization-test', and 'narenlk\_csv\_dynamodb'. Each row shows the AWS service associated with the role and the last activity date.

Just follow this, under **Services** → **select IAM** → **Roles** → **Create Role**

Now select AWS Service → EC2 (under common use case) → next:

## Permission

Select type of trusted entity

**AWS service**  
EC2, Lambda and others

**Another AWS account**  
Belonging to you or 3rd party

**Web identity**  
Cognito or any OpenID provider

**SAML 2.0 federation**  
Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

**Common use cases**

**EC2**  
Allows EC2 instances to call AWS services on your behalf.

**Lambda**  
Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway	CodeBuild	EMR	IoT SiteWise	RDS
AWS Backup	CodeDeploy	EMR Containers	IoT Things Graph	Redshift
AWS Chatbot	CodeGuru	ElastiCache	KMS	Rekognition

\* Required      [Cancel](#)      [Next: Permissions](#)

choose\_use\_cases

Alright now we have to attach the policy which we created earlier with this Role, as shown below,

▼ Attach permissions policies

Choose one or more policies to attach to your new role.

[Create policy](#) 

Filter policies		<input type="text" value="narenlt"/>	Showing 1 result
	Policy name		Used as
<input checked="" type="checkbox"/>	narenltkGrafanaPolicy		None

attach\_permission

Once the Policy has been attached we gotta review it, where ought to update the Role Name and Description etc.. and once we are done with this we can select Create Role.

Create role

Review

Provide the required information below and review this role before you create it.

**Role name\*** narenitkGrafanaRole  
Use alphanumeric and '+,-,@-\_ characters. Maximum 64 characters.

**Role description** Allows EC2 instances to call AWS services on behalf of almighty narenitk.  
Maximum 1000 characters. Use alphanumeric and '+,-,@-\_ characters.

**Trusted entities** AWS service: ec2.amazonaws.com

**Policies** narenitkGrafanaPolicy 

**Permissions boundary** Permissions boundary is not set

\* Required Cancel Previous **Create role**

**review\_role**

Once that is done we need to verify whether the role which we created is there or not,

The screenshot shows the AWS IAM Roles page. At the top, a green success message box says "The role narenltkGrafanaRole has been created." Below the message are two buttons: "Create role" (blue) and "Delete role" (grey). A search bar contains the text "narenltk". The main table lists roles with the following data:

Role name	Trusted entities	Last activity
narenltk_csv_dynamodb	AWS service: lambda	156 days
narenltkGrafanaRole	AWS service: ec2	None

## Step 6: Creating a User

First, we have created the Policy then Role and then attached the Policy with the Role and now we are creating the User so that we can attach it to the EC2 Instance.

The screenshot shows the AWS IAM Users page. On the left, a sidebar menu under "Identity and Access Management (IAM)" includes "Dashboard", "Access management" (with "User groups", "Users" (highlighted in orange), "Roles", "Policies", "Identity providers", and "Account settings"), and "Access reports". The main area has buttons "Add user" (blue) and "Delete user" (red). A search bar says "Find users by username or access key". The table below shows no users: "User name" (dropdown), "Groups", "Access key age", and "Password age". A message at the bottom says "There are no IAM users. [Learn more](#)".

create\_iam\_users

Now select the Add User.

## Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\*

[+ Add another user](#)

## Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type\*  **Programmatic access**

Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

**AWS Management Console access**

Enables a **password** that allows users to sign-in to the AWS Management Console.

\* Required

[Cancel](#)

[Next: Permissions](#)

**update\_user\_details**

Give the user name.

Kindly make sure that you select the programmatic access in the Access Types under select AWS access type. Which will give you the details of the Access Key and Secret Key. Make sure that you make a note of it as it will be generated only once and if you forget it then you need to create it again.

Then select Next Permission.

▼ Set permissions

The screenshot shows the 'Set permissions' section of the AWS IAM console. It features three main options: 'Add user to group', 'Copy permissions from existing user', and 'Attach existing policies directly'. The third option is highlighted with a blue border. Below these are buttons for 'Create policy' and a refresh icon. A search bar labeled 'Filter policies' and a dropdown menu are also present. A table lists a single policy named 'narenltkGrafanaPolicy' with details: Type 'Customer managed' and Used as 'Permissions policy (1)'. The table has columns for Policy name, Type, and Used as.

Policy name	Type	Used as
narenltkGrafanaPolicy	Customer managed	Permissions policy (1)

attach\_grafana\_policy

Now Set Permission, if you have already created any group means select add user to group or else proceed with Attach existing policies directly → search for the Policy which we created in the last step and proceed.

A green success message box is displayed, indicating that the user was successfully created. It provides instructions for signing in to the AWS Management Console and includes a link: <https://352240879602.signin.aws.amazon.com/console>. Below the message is a 'Download .csv' button. A table lists the user 'narenltkGrafanaUser' with columns for User, Access key ID, and Secret access key. The Access key ID is shown as AKIAVEAZRI7ZCGSD4EUP, and the Secret access key is shown as a masked value '\*\*\*\*\* Show'.

User	Access key ID	Secret access key
narenltkGrafanaUser	AKIAVEAZRI7ZCGSD4EUP	***** Show

grafana\_user\_Cred

Now here you can see the Access Key ID and Secret access key, which I was talking about earlier.

Yeah, you have to save it since it will be displayed only once just like the proverb “**The Golden Words are not repeated**”.

I guess now we are all set to go.... Now all that's left is just attach the User with the EC2 instance which we have.

## Step 7: Attach the User to EC2 Instance

Here I do expect you all to have your EC2 instance ready i.e. it should have been started.

The screenshot shows the AWS EC2 Instances page with one instance listed:

Name	Instance ID	Instance state	Instance type	Status
narenltk	i-0741864d1c5c44f8f	Running	t2.micro	Up to date

Below the table, the instance details are shown:

Instance: i-0741864d1c5c44f8f (narenltk)

Details | Security | Networking | Storage | Status checks

Change security groups | Get Windows password | Modify IAM role

Instance summary | Info

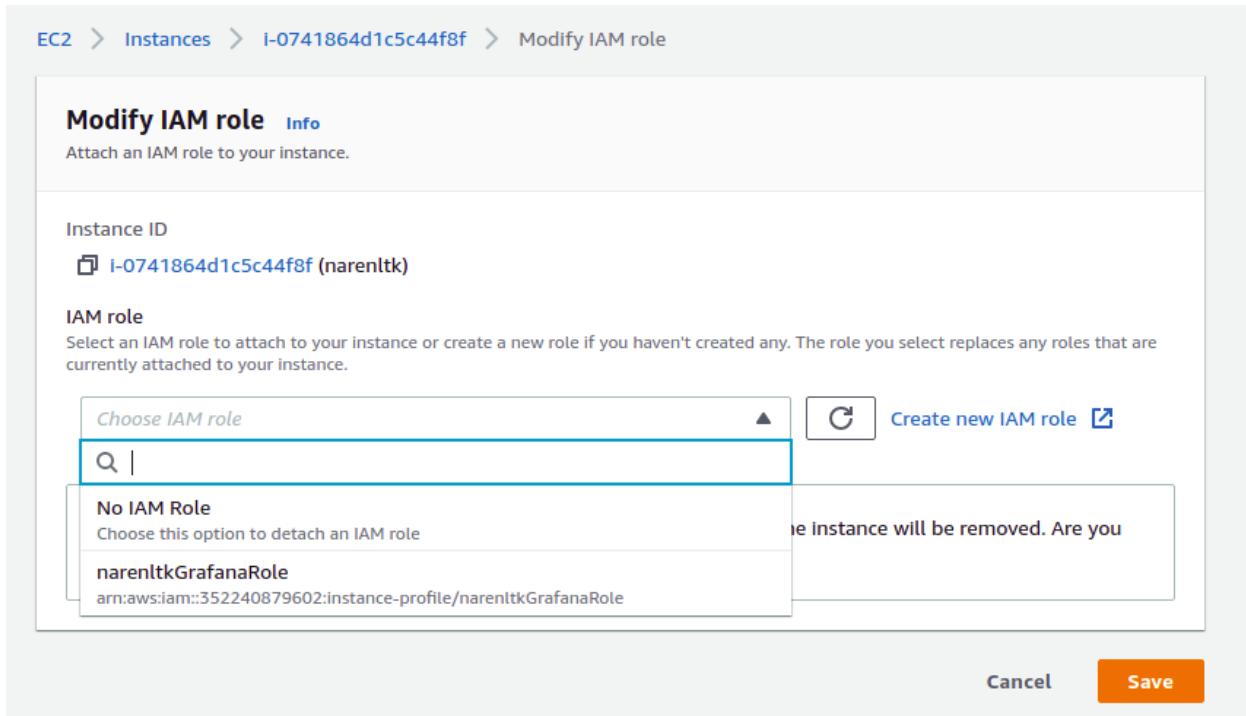
Instance ID i-0741864d1c5c44f8f (narenltk)	Public IPv4 address 54.166.146.93   open address	Private IPv4 addresses 172.31.47.4
Instance state Running	Public IPv4 DNS ec2-54-166-146-93.compute-attach_iam_role	Private IPv4 DNS ip-172-31-47-4.ec2.internal

Now select the instance for which you wanted to update the IAM User and click on

**Action → Security → Modify IAM Role.** (This is the latest since AWS has been keep on updating)

If you have been referring to the Old videos on YouTube or else some old Medium Blogs then you would have been navigated to

**Action → Instance Settings → Attach / Replace IAM Role.**



Now you gotta search for the Role which you have created and select that and save it.

Now you ought to verify that too, so kindly check the security group settings of the EC2 instance as shown below,

The screenshot shows the AWS EC2 Instances page. At the top, a green banner displays the message: "Successfully attached narenltkGrafanaRole to Instance i-0741864d1c5c44f8f". Below the banner, the "Instances (1/1)" section lists a single instance named "narenltk" with the ID "i-0741864d1c5c44f8f". The instance is "Running" and has an "t2.micro" instance type. It has passed 2/2 checks and has no alarms. The "Availability Zone" is "us-east-1c". Below this, the "Instance: i-0741864d1c5c44f8f (narenltk)" details page is shown. The "Security" tab is selected, showing the attached IAM Role "narenltkGrafanaRole". Other tabs include Details, Networking, Storage, Status checks, Monitoring, and Tags. The "Security details" section also lists the Owner ID "352240879602" and the Launch time "Sun May 02 2021 12:00:10 GMT+0530 (India Standard Time)".

You can verify the IAM Role that is attached to the Security Group.

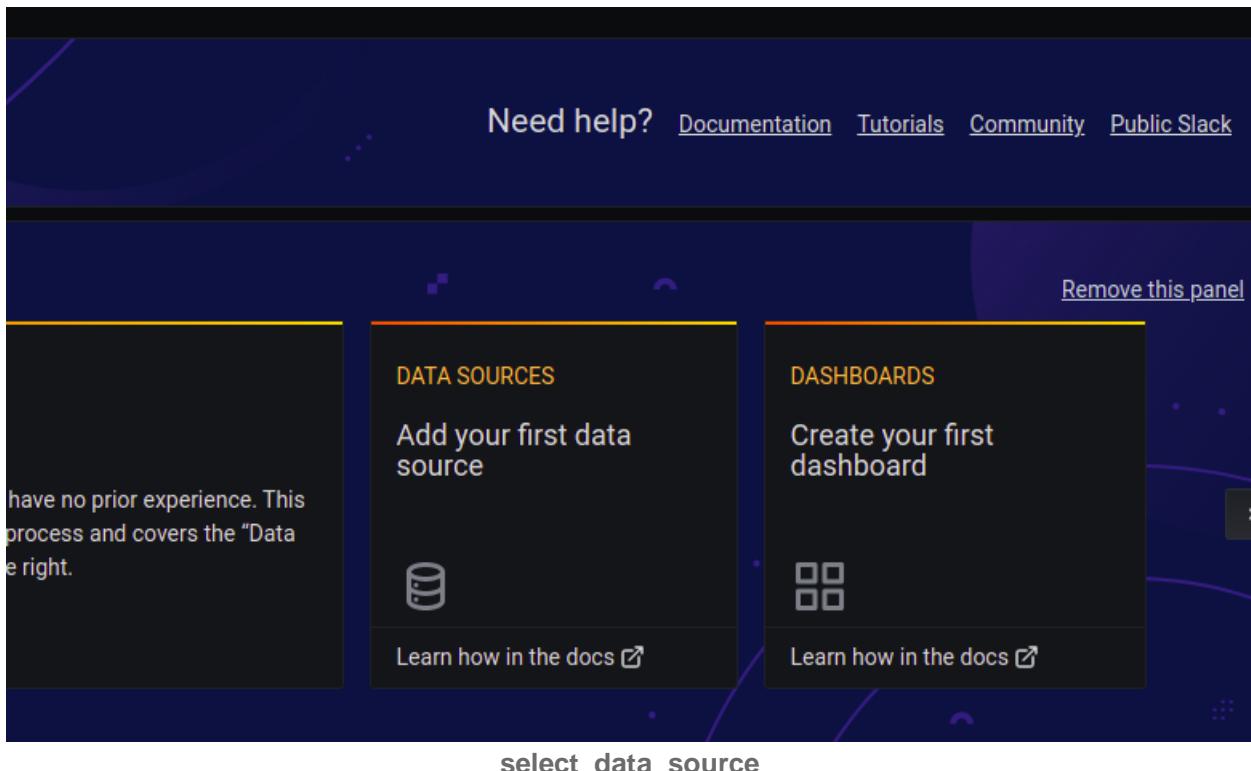
## Step 8: Configure Grafana with AWS Cloud Watch

Now log into to your Grafana Dashboard, with uname and pwd which you have configured,

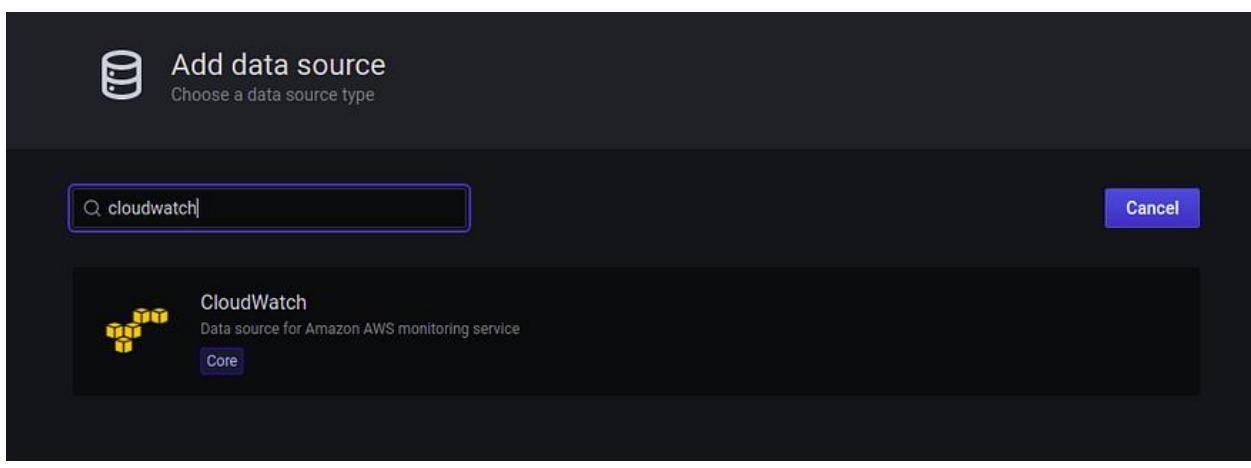
***Note:Default uname and pwd is given below***

***Uname → admin***

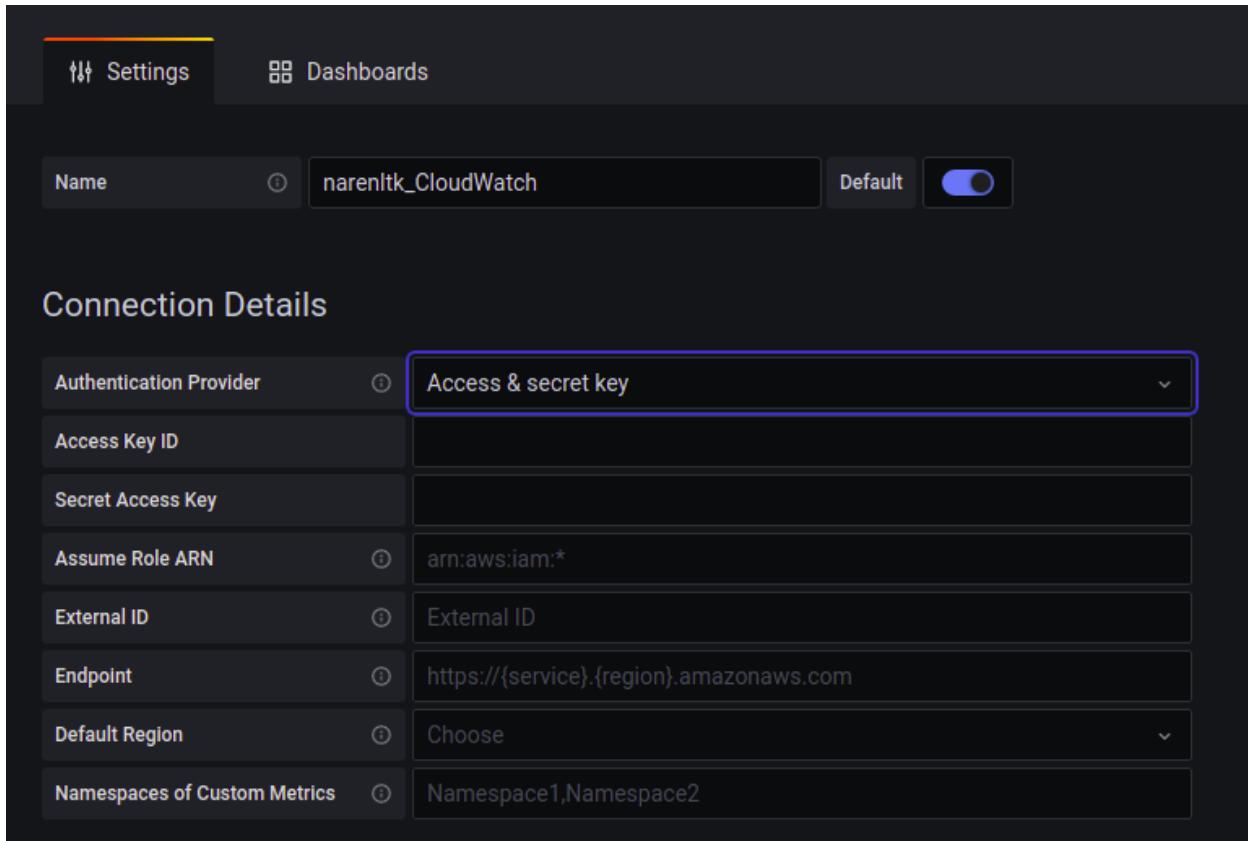
**Pwd → admin**



Here you ought to see a dashboard panel like shown above and kindly select the Data Source, i.e Add your first data source.



Now search for the Cloud Watch in the search bar, then select it to configure the credential details.



Here you gotta give your desired name, then under Authentication Provider select the Access & Secret Key.

Now given the details of the Access Key and Secret Access Key alone with the Default Region.

Then click on Save and Test and you should get the following as the output.

The screenshot shows the 'Connection Details' configuration page in Grafana. It lists various AWS authentication parameters:

Authentication Provider	Access & secret key
Access Key ID	Configured
Secret Access Key	Configured
Assume Role ARN	arn:aws:iam:*
External ID	External ID
Endpoint	https://(service).(region).amazonaws.com
Default Region	us-east-1
Namespaces of Custom Metrics	Namespace1, Namespace2

A green success message at the bottom states: ✓ Data source is working.

`verify_updated_details_correct`

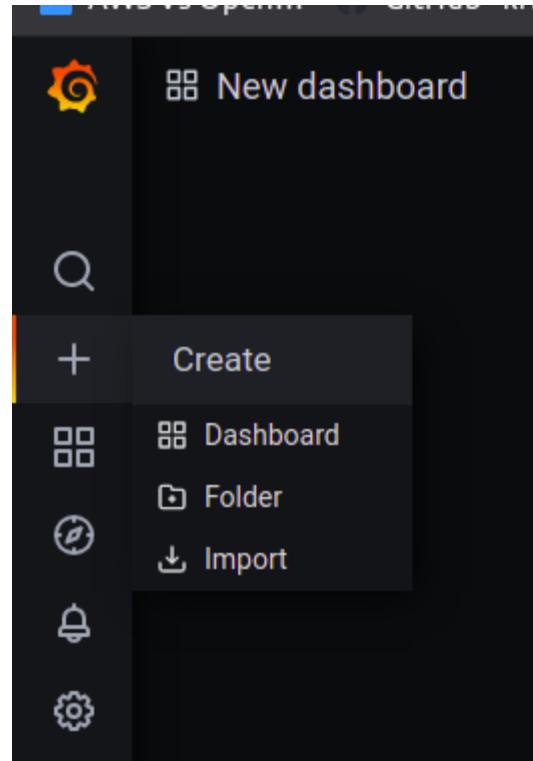
If in any place you find any error which you wanna track then, go to the EC2 instance Terminal and give the following **command to check the log reports,**

```
cd /var/log/grafana/lsvi grafana.log (or else) nano grafana.log
```

This will give you all the details of the error which you have faced.

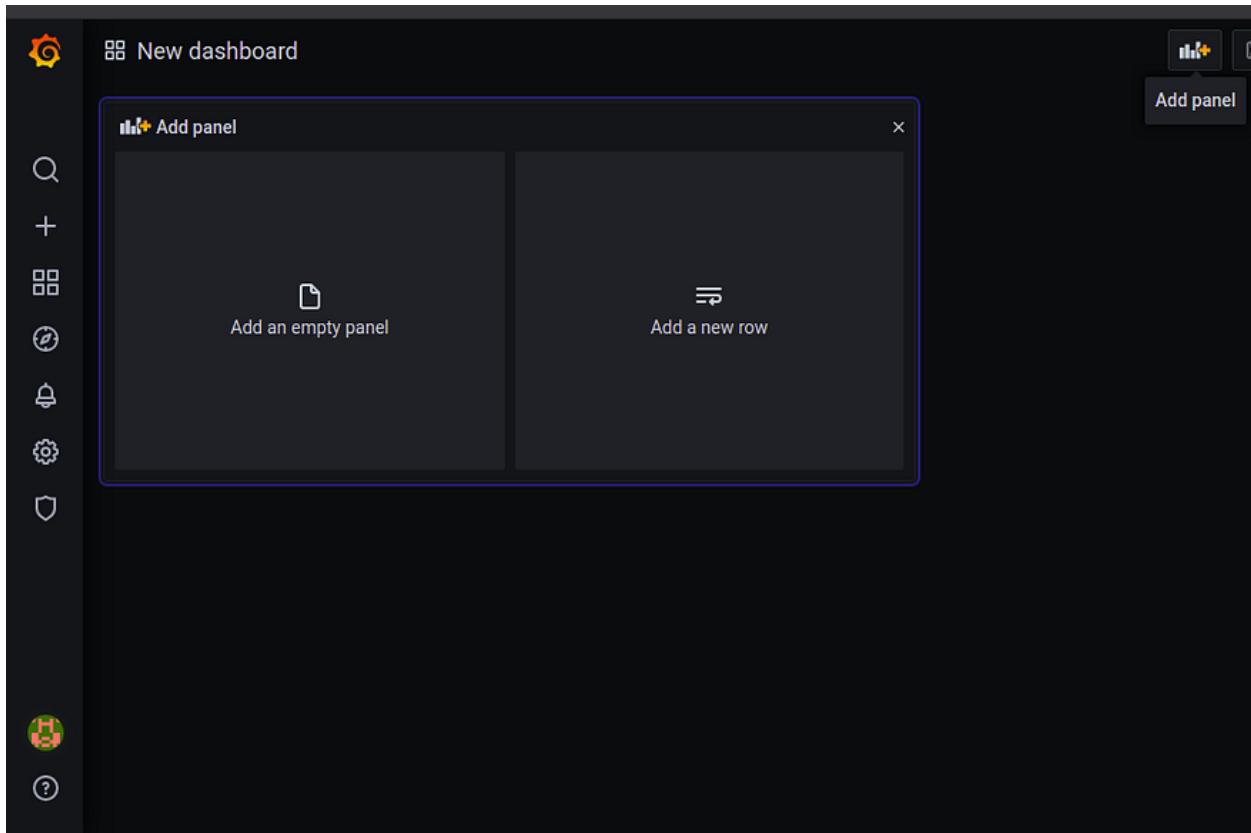
## Step 9: Create your beautiful Dashboard

Now you need to select the “ + → Dashboard “



`click_dashboard_for_making_panel`

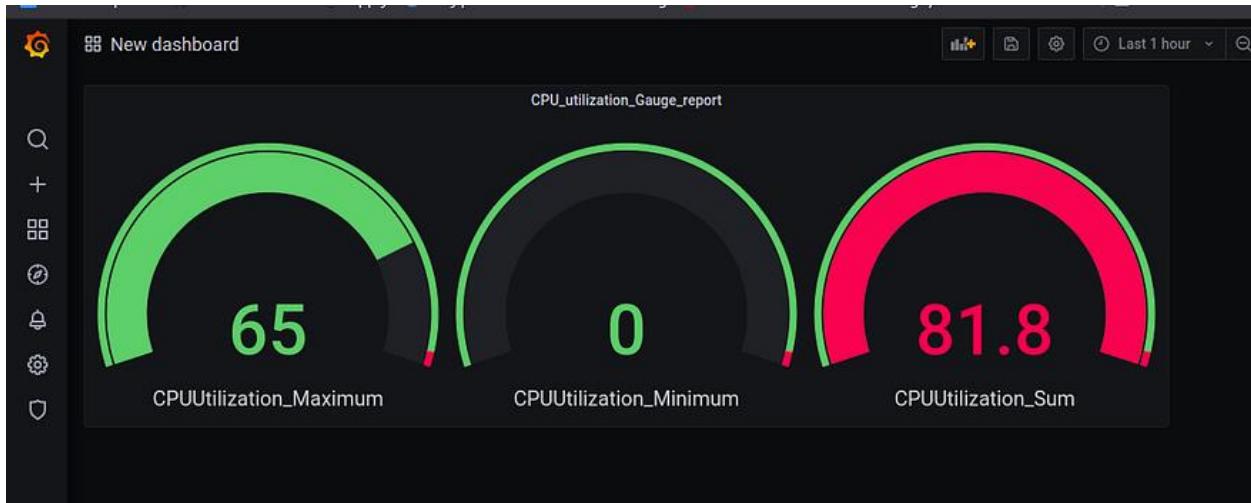
Now you can either select the Add Panel in the top right corner as sometimes panel will be created created with Add Empty Panel and Add Row as shown below,



`select _add_panel`

Then all the grafana query I will let you guys do it since it is easy though.

Here I have done an example with the CPU utilization.



## Conclusion:

Certainly! Integrating AWS CloudWatch with Grafana enables seamless visualization and monitoring of AWS resources. Grafana's dynamic dashboard creation offers flexibility in analyzing various metrics from AWS services. Whether you choose to use the Amazon CloudWatch data source, AWS Observability CloudWatch metrics with the AWS Observability app, Grafana Alloy with the CloudWatch exporter, or the yet-another-cloudwatch-exporter (YACE), each option leverages Amazon CloudWatch APIs to make the data available in Grafana. Remember that Grafana Cloud provides an easy way to start observing your AWS environment. Good luck with your case study!

